

Exercício - FURPS

Classificando Atributos de Qualidade

Nome: **Barbara Espanhol**

RA: 321119046

Nome: **Isamara Lorrane**

RA: 320137186

Nome: **Loyanne Carvalho**

RA: 320110975

Nome: **Lucas Etrusco**

RA: 320129659

Nome: **Thales Rocha**

RA: 321111555

Nome:

RA:

- **Atividade em Grupo:** 6 participantes
- Escolham um representante do grupo para criar um repositório no Github, e, adicionar os outros integrantes do grupo como colaboradores <https://docs.github.com/pt/account-and-profile/setting-up-and-managing-your-github-user-account/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository>
- Atualize a planilha [Grupos - Gerência de Qualidade](#)
- Após terminar a atividade:
 - Adicione esse arquivo no formato PDF no repositório;
 - Cada integrante do grupo, poste o arquivo PDF no ulife.

O documento de *Especificação Suplementar de Requisitos* captura os requisitos de sistema que não são capturados imediatamente nos casos de uso do modelo de casos de uso. Entre os requisitos estão incluídos os seguintes atributos de qualidade do sistema: usabilidade, confiabilidade, desempenho e suportabilidade. Tais requisitos fazem parte da FURPS, que é um modelo de classificação de atributos de qualidade de software, desenvolvido na Hewlett-Packard (HP) e publicado pela primeira vez por Grady e Caswell.

Considerando um sistema qualquer, preencha cada uma das seções abaixo do documento de Especificação Suplementar de Requisitos referente à FURPS. A atividade deverá ser feita em sala, com equipes de 6 alunos até o final da aula.

Nome do Sistema

Uber

Descreva resumidamente o sistema e suas principais funcionalidades

Aplicativo para transporte de pessoas e mercadorias. A sua principal funcionalidade é solicitar um motorista para realizar uma corrida com pessoas ou objetos.

1. Funcionalidade

Descreva abaixo os requisitos funcionais do seu sistema, colocando o nome da funcionalidade e a descrição dela, por exemplo:

- **E-mail:** fornecer serviços que permitam que os usuários enviem e recebam mensagens;
- **Ajuda Online:** disponibilizar ajuda online para os usuários;
- **Segurança:** Proporcionar serviços para proteção de acesso a determinados recursos ou informações.

- **Adicionar endereços:** Deixar endereços muito utilizados salvos para futuras corridas.
- **Iniciar corrida:** Ao adicionar endereço de saída e chegada, o motorista é solicitado para iniciar uma corrida.
- **Categorias de transporte:** O usuário pode escolher o tipo de transporte de acordo com o que ele precisa. (ex: uber flash para entrega de mercadorias)
- **Pagamento:** O usuário pode escolher a forma de pagamento (dinheiro ou cartão).
- **Avaliação:** O usuário pode avaliar o motorista de acordo com satisfação de cada corrida

- realizada.
- Compartilhamento de rota: O usuário pode compartilhar a corrida com outra pessoa para se sentir mais seguro.
- Perfis: O usuário pode verificar a veracidade dos dados do motorista como, placa, foto, modelo do carro, avaliação dada pelos outros usuários.
- Agendamento de viagem: O usuário pode agendar uma corrida para o mesmo dia, no horário que melhor te atende.
- Chat: O usuário pode entrar em contato com o motorista caso aconteça um imprevisto (ex: avisar que está com um animal de estimação, que esqueceu algo no carro, etc.).

2. Usabilidade

Descreva nesta seção todos os requisitos de qualidade relacionados a usabilidade, tais como: facilidade de uso, facilidade de aprendizado, padrões de usabilidade e localização; Por exemplo:

- Tempo de treinamento necessário para que usuários comuns ou avançados se tornem produtivos em operações específicas no sistema;
- Especifique períodos de tempo mensuráveis para tarefas típicas no sistema;
- UI/UX Design Patterns.

Os requisitos de usabilidade podem incluir as seguintes subcategorias: fatores humanos, estética, consistência na interface com o usuário, ajuda on-line sensível ao contexto, assistentes e agentes, documentação do usuário e materiais de treinamento.

Facilidade de uso, layout de fácil entendimento, facilidade no aprendizado, padrões de usabilidade, praticidade de utilização e versatilidade de pagamento e acessibilidade.

3. Confiabilidade

Os requisitos de confiabilidade a serem considerados são: frequência e gravidade de falha, capacidade de recuperação, possibilidade de previsão, precisão e tempo médio entre falhas (MTBF). Por exemplo:

- A. Disponibilidade:** especifique a porcentagem de tempo disponível (xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.
- B. Tempo Médio entre Falhas (MTBF):** normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.

- C. Tempo Médio para Reparo (MTTR):** quanto tempo o sistema poderá ficar sem funcionar após uma falha?
- D. Exatidão:** especifique a precisão (resolução) e exatidão (através de algum padrão conhecido) necessárias na saída do sistema.
- E. Taxa máxima de erros ou defeitos:** geralmente expressa em termos de erros / KLOC (thousands of lines of code, milhares de linhas de código) ou de erros / ponto de função.
- F. Taxa de erros ou defeitos (categorizada em termos de erros de pouca, média ou muita importância):** os requisitos devem definir o que se entende por erro “crítico”(por exemplo, perda total de dados ou total incapacidade de usar determinadas partes da funcionalidade do sistema).

Descreva abaixo os requisitos de confiabilidade para seu sistema referente aos itens A, B e C da lista acima:

Disponibilidade: A Uber fica disponível 24h por dia, desde que o usuário esteja conectado a uma rede de internet, a manutenção é realizada no mesmo período de uso do usuário, todas as formas de pagamento ficam disponíveis 24h.

Tempo médio entre falhas: 2 dias

Tempo médio para reparo: menor que 60 minutos.

4. Desempenho

Descreva as características de desempenho do seu sistema, tais como: velocidade, eficiência, disponibilidade, precisão, produtividade, tempo de resposta, tempo de recuperação e uso de recursos. Por exemplo:

- **Tempo de resposta de uma transação:** tempos médio e máximo;
- **Taxa de transferência:** quantidade de transações por segundo;
- **Capacidade:** o número de clientes ou de transações que o sistema pode acomodar;
- **Modos de degradação:** o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira;
- **Uso de recursos:** memória, disco, comunicações, etc

Tempo de resposta de uma transação: tempos médio e máximo – 3 segundos.

Capacidade: o número de clientes ou de transações que o sistema pode acomodar – 1000 pessoas por segundo.

Modos de degradação: o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira;

Uso de recursos: memória, disco, comunicações, etc – Celular com GPS, mínimo 2gb de ram, 20mb livres no hd, acesso à internet.

5. Suportabilidade

Descreva todos os requisitos que aprimorarão a *suportabilidade* ou *manutenibilidade* do seu sistema, tais como: padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção. Ainda, os requisitos de suportabilidade podem incluir as possibilidades de teste, adaptação, manutenção, compatibilidade, configuração, serviço, instalação e localização (internacionalização)

Linguagem: Javascript.
Design pattern: Robot Pattern.
Organização: Modularização
Biblioteca: Nodejs-npm.